



Laurea Intranetin testaus



Kuosma, Riikka

2010 Kerava

Laurea-ammattikorkeakoulu
Laurea Kerava

Laurea Intranetin testaus

Riikka Kuosma
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Lokakuu, 2010

Riikka Kuosma

Laurea Intranetin testaus

Vuosi	2010	Sivumäärä	47
-------	------	-----------	----

Tämän opinnäytetyön aiheena on Laurea Intranetin käyttöönotto ja siihen liittyvän hyväksymistestauksen suunnittelu ja toteutus. Laurea on uutta osaamista tuottava ammattikorkeakoulu, jonka keskeisessä asemassa on työelämäläheisyys. Ennestään Laurealla oli käytössä vain Internet-sivusto. Sisäinen viestintä on yksi keskeisimmistä toiminnan kehittämisalueista, joten intranetin koettiin olevan tarpeellinen työväline sisäisen viestinnän ja suuren tietomäärän jakamiseksi.

Opinnäytetyö sisältää teoria osuuden, kuvauksen Laurea Intranet-projektista sekä omia ajatuksia koko projektista. Teoria osuudessa käsitellään seuraavanlaisia asioita: intranetiä, vaatimusmäärittelyä, vaatimustenhallintaa ja testausta. Projektin kuvauksessa käydään läpi projekti kokonaisuudessaan ja arvioidaan hyväksymistestauksen onnistumista. Järjestelmän testaukset toteutettiin rakentamalla testauslomake, joiden pohjalta testituloksia analysoitiin. Lisäksi testaus suoritettiin järjestelmätoimittajan omilla testauslomakkeilla.

Testauslomakkeista saamiemme tuloksien perusteella järjestelmä oli ollut hyvin keskeneräinen ja puutteellinen vielä testausvaiheessa, joten järjestelmä testaaminen ei onnistunut suunnitelmien mukaisesti.

Lopuksi voidaan todeta, että projekti ei onnistunut odotetulla tavalla ja nykyinen Intranet ei ole käyttäjäystävällinen kuten alun perin oli suunniteltu. Tällä hetkellä Laureassa on aloitettu uusi Intranet-projekti ja uusi Intranet-palvelu tulee korvaamaan nykyisen järjestelmän.

Asiasanat: vaatimusmäärittely, intranet, testaus

Riikka Kuosma

Intranet Testing. CASE: Laurea University of Applied Sciences

Year	2010	Pages	47
------	------	-------	----

This Bachelor's thesis focuses on planning and implementing of acceptance test for Laurea Intranet. Laurea is a University of Applied Science, which creates new competences and carries out professionally orientated education. Internal communication is one of the main development areas, which is why the intranet is seen as a necessary tool for sharing internal communication and a large amount of data.

The thesis includes of a theory section and an empirical section that deals with Laurea's Intranet project. The theory section discusses intranet, requirements specification, requirements management and testing. The empirical section deals with whole Laurea Intranet project and success of acceptance test. Test results were analyzed by put on acceptance test.

The test results indicate that the software was incomplete. The acceptance test was moved forward by two weeks in order for the software to be ready for the acceptance test's thorough implementation. The software was not input enough data. More than half of the tested items gave an erroneous result.

In conclusion the project was not as successful as expected. According to the feedback received from the testers the test forms was even better. The current intranet was not as usable as originally planned. Laurea started a new intranet project, which replaced the current intranet.

Key words: requirements specification, intranet, testing

Sisällys

1	Johdanto	6
2	Projekti Laurean Intranet-palvelu	6
2.1	Tavoite	6
2.2	Lähtökohdat ja tausta	7
2.3	Aikataulu	7
2.4	Organisaatio, ohjaus ja työnjako	7
2.5	Seuranta ja valvonta	8
2.6	Hallintamenettelyt ja dokumentit	8
3	Intranet	9
4	Vaatimusmäärittely	10
5	Vaatimusten hallinta	12
6	Testaus	14
7	Laurean Intranetin testaus	17
8	Arviointi	19
9	Laurea Intranet tänään	21
	Lähteet	24
	Kuvat	25
	Liitteet	26

1 Johdanto

Opinnäytetyöni aiheena on Laurean Intranetin hyväksymistestauksen suunnittelu ja toteutus. Tulin mukaan tähän projektiin sen loppuvaiheessa syksyllä 2007, jolloin tarkoituksena oli suunnitella ja toteuttaa hyväksymistestaus tulevalle järjestelmälle.

Laurea on uutta osaamista tuottava ammattikorkeakoulu, jonka keskeisessä asemassa on työelämäläheisyys. Laureassa on noin 8000 opiskelijaa, joista noin 1200 on aikuisopiskelijoita. Henkilöstöä Laureassa on noin 500. Laurealla on ennestään pelkkä Internet-sivusto. Sisäinen viestintä on yksi keskeisimmistä toiminnan kehittämisalueista tällä hetkellä. Tärkeimmäksi työvälineeksi nousee ehdottomasti intranet, jonka on tarkoitus olla käyttövalmis keväällä 2008.

Testauksen ja testauslomakkeiden suunnittelussa käytin apuna vaatimusmäärittelyä ja projektisuunnitelmaa. Järjestelmän ulkoasun ja käyttöliittymän hahmottamisen avuksi sain Adage Oy:n tuottamat käyttöliittymämallit (Liite 4). Testauslomakkeet oli määrä saada valmiiksi tammikuussa 2008, jonka jälkeen aloittaisivat järjestelmän testaamisen suunnittelemini lomakkeiden pohjalta. Lisäksi julkaisujärjestelmää testattiin toimittajan omaa testauslomaketta apuna käyttäen.

Opinnäytetyössäni käyn läpi testauksen perusteita kirjallisuuden avulla sekä projektin vaihe vaiheelta omia kokemuksieni perusteella. Lopuksi analysoin saamiani testaustuloksia ja projektin onnistumista.

2 Projekti Laurean Intranet-palvelu

2.1 Tavoite

Projektin tavoitteena on suunnitella ja toteuttaa Laurean intranet-sivusto. Palvelun keskeisiä toiminnallisuuksia ovat hyvät hallinta- ja ylläpitotoiminnot, laajat loppukäyttäjätöiminnot ja toimivat rajapinnat ulkopuolisten järjestelmien yhteensovittamiseen. Palvelu toteutetaan Livelink WCM Server -julkaisujärjestelmällä, joka Laurealla on käytössä ennestään ulkoisen sivuston hallinnassa. (Suur-Askola 2007, 5.)

2.2 Lähtökohdat ja tausta

Laurea yksi suurimpia ammattikorkeakouluja Suomessa. Laureassa on opiskelijoita noin 8000, joista n. 1200 aikuiskoulutuksessa ja henkilöstöä n. 500. Sisäinen viestintä on yksi keskeinen toiminnan kehittämisen alue tällä hetkellä. Tärkeimmäksi välineeksi kyseisellä alueella on nostettu intranet, jonka on suunniteltu olevan toiminnassa vuoden 2008 keväällä. (Suur-Askola 2007, 5.)

2.3 Aikataulu

Projekti jakaantuu määrittely-, toteutus- ja käyttöönottovaiheeseen. Projekti aloitettiin 15.8.2007. Järjestelmän määrittelyvaihe aloitettiin ensimmäisellä työpajalla 23.8.2007. Toteutusvaiheen oli määrä alkaa 3.10., kun määrittelyvaihe oli hyväksytty. Käyttöönotto vaihe voitiin aloittaa, kun toteutusvaihe on valmistunut ja tarvittava laitteistot ja ohjelmistot on toimitettu ja asennettu järjestelmän käyttöympäristöön. Hyväksymistestaus oli tarkoitus aloittaa tammikuussa 2008. (Suur-Askola 2007, 6-9.)

2.4 Organisaatio, ohjaus ja työnjako

Projektia varten muodostettiin kaksi projektiryhmää. Projektiryhmät koostuivat Laurean ja Onestan projektiryhmistä. Lisäksi projektiin nimettiin mukaan ohjausryhmä, jonka tarkoituksena oli tehdä päätökset sellaisista projektiin vaikuttavista tekijöistä, jotka vaikuttavat projektin kustannuksiin, sisällön muutoksiin tai aikataulujen muutoksiin. (Suur-Askola 2007, 10.)

Toimittajan eli Onestan projektiryhmään kuuluivat projektipäällikkö Kimmo Suur-Askola, jonka vastuualueeseen kuului projektin hallinta ja vaatimusmäärittely. Käytettävyyssiantuntijoiden vastuulla oli rakenne- ja käyttöliittymä suunnittelu ja HTML-taitto. Teknisen projektipäällikön ja arkkitehdin vastuulla oli toiminnallinen vaatimusmäärittely ja teknisen toteutuksen käyttöönotto. Ohjelmistosuunnittelijan vastuulla oli äsken mainitut vastuut. Mukana oli visuaalinen suunnittelija. (Suur-Askola 2007, 10.)

Asiakkaan eli Laurean projektiryhmä koostui seuraavista henkilöistä. Projektipäällikkönä toimi Merja Jalava, jonka vastuulla oli projektin hallinta, määrittely ja hyväksymistestaus. Teknisen projektipäällikön vastuulla oli tekninen IT-hallinto asiakkaan puolelta. Projektsihteerin tehtävänä olivat kokousten dokumentointi ja yleiset asiat. Tietoympäristöinformaatikko vastasi tiedon löytyvyydestä. Graafinen suunnittelija vastasi visuaalisesta suunnittelusta järjestelmän osalta. Viestintäassistentti huolehti projektissa viestinnän tarpeista, kehityspäällikkö puolestaan opintojen ohjauksesta. (Suur-Askola 2007, 11.)

2.5 Seuranta ja valvonta

Projektin seuranta ja valvonta oli päätetty toteuttaa kokouksien eli projektipalaverien avulla, joihin osallistuivat Luran ja Onestan projektipäälliköt ja tarvittavat henkilöt Laurean ja Onestan projektiryhmistä. Ohjausryhmän kokouksiin osallistuivat vain ohjausryhmään kuuluvat jäsenet. Lisäksi sovittiin, että jokaisesta toimittajan ja asiakkaan välisestä kokouksesta vaaditaan esityslista, joka toimitetaan etukäteen sähköisessä muodossa kokoukseen osallistuville. (Suur-Askola 2007, 13.)

Ohjausryhmän kokouksesta vastaa asiakkaan projektipäällikkö yhdessä toimittajan projektipäällikön kanssa. Ohjausryhmän kokouksista tehtiin pöytäkirja, johon merkittiin tehdyt päätökset ja käsittelemättömät tai avoimeksi jääneet asiat. Jakelu tapahtui sähköisesti ohjausryhmän ja projektiryhmän jäsenille. Toimittajan projektipäällikön vastuulle jäi projektin etenemisen raportointi asiakkaan projektipäällikölle vähintään kerran viikossa koko projektin ajan. (Suur-Askola 2007, 13-14.)

Projektin lopussa toimittajan projektipäällikkö kirjoitti loppuarvioinnin, jossa analysoitiin asioita, jotka ovat vaikuttaneet projektin onnistumiseen sekä asioita, jotka olisi voitu tehdä paremmin tai parantaa. Loppuarvioinnissa lähteinä käytettiin projektin aikana syntyneitä dokumentointia, jäsenille tehtyä kyselylomaketta sekä jäsenten ja projektipäällikön ajatuksia ja muistikuvia. (Suur-Askola 2007, 14.)

2.6 Hallintamenettelyt ja dokumentit

Projektin aikana tuotettiin useita erilaisia dokumentteja esim. projektisuunnitelma, kokouspöytäkirjat, vaatimusmäärittely, käyttöliittymämallit, testaussuunnitelma, testiraportit, tekninen dokumentaatio ja ylläpito-ohjeistus. (Suur-Askola 2007, 14.)

Riskejä pyritään tunnistamaan, analysoimaan ja aktiivisesti hallitsemaan koko projektin ajan. Jokainen projekti- ja ohjausryhmän jäsen oli velvoitettu ilmoittamaan mahdollisista riskeistä, mitä projektin aikana tulee esiin. Riskienhallinta kokonaisuudessaan oli ohjausryhmän vastuulla. (Suur-Askola 2007, 14.)

Muutoksella tarkoitetaan tässä yhteydessä uutta vaatimusta projektin sisältöön tai aikatauluun. Muutokset piti hyväksyttää projektin ohjausryhmällä ja jokaisesta muutoksesta kirjattiin seuraavan asiat: kirjauspäivämäärä, syy, kuvaus, vaikutukset ja työmäärä sekä hyväksymis- tai hylkäyspäätös, päätöksen tekijät ja päivämäärä. (Suur-Askola 2007, 15.)

Projektin tuotoksena syntyneistä dokumenteista ainoastaan kokousten pöytäkirjoista ei pidetty yllä versioita, muu dokumentaatio tallennettiin omaksi versiokseen.

Projektin vaadittava ohjelmakoodi toteutettiin ”Java Server Pages” -tekniikalla. Tässä projektissa ohjelmakoodia koskevaa versionhallinnan apuna käytettiin julkaisujärjestelmän sisäistä versionhallintaa ja versiolokia. Toiminnallisten ohjelmakomponenttien eli Java Bean -tekniikalla toteutettuja ja ”JSP tag library” - tekniikkaan liitettyjä ohjelmakoodien versionhallintaan käytettiin SVN-versionhallintaohjelmistoa. (Suur-Askola 2007, 15.)

Laadunhallinnallisena toimenpiteenä järjestelmälle suoritettiin moduuli-, integraatio-, järjestelmä- ja suorituskykytestaus ennen järjestelmän toimittamista asiakkaalle. Lisäksi järjestelmälle suoritettiin hyväksymistestaus asiakkaan toimesta. Projektin aikana tuotetut dokumentit toimivat yhtenä osana laadunvarmistusta ja ohjausryhmä katselmoi kaikki keskeiset dokumentit. Projektissa pyritään noudattamaan julkisten verkkopalveluiden laatukriteereitä. (Suur-Askola 2007, 15.)

Projektin tuotokset hyväksyy ohjausryhmä ja hyväksymisestä tehdään kirjallinen merkintä. Projektin lopputuloksen hyväksyy ohjausryhmä, kun projektin tuotokset on hyväksytty.

Järjestelmän käytettävyydestä osalta sovittiin, että Laurea voi pyytää arviota ulkopuoliselta asiantuntijalta kun järjestelmä on valmis. Käytettävyydestä ei kuulunut tähän projektiin, eikä siis vaikuttanut sen hyväksymiseen. Käytettävyydestä aikana havaitut muutostarpeet toteutettaisiin jatkokehitystyönä. (Suur-Askola 2007, 16.)

3 Intranet

Intranet eli sisäinen verkko perustuu samaan tekniikkaan kuin maailman laajuinen Internetkin. Intranet on rajoitettu osa Internetiä, jossa käytetään samoja protokollia ja toimintoja kuin Internetissä. Yrityksen intranet erotetaan Internetistä käyttämällä verkkokomponentteja, tietokoneita ja ohjelmistoja. Verkkokomponentit, tietokoneet ja ohjelmistot muodostavat palomuurin, jonka avulla vain yrityksen työntekijät pääsevät hyödyntämään Internetin resursseja, mutta ulkopuoliset eivät pääse Internetin kautta yrityksen intranetiin. Yrityksen sisäisessä käytössä oleva intranet on tapa jakaa tietoa ja muita resursseja työntekijöiden kesken. Intranetin avulla voidaan välittää mm. tekstejä, grafiikkaa, kuvia, videoita ja ääntä. Intranetissa voidaan tarjota virtuaalisia työtiloja tiimien käyttöön, joiden avulla kokousten pitäminen mahdollistuu maantieteellisistä etäisyyksistä huolimatta. Intranet-sivu näyttää ulkoasullisesti aivan samalta kuin mikä tahansa Internet-sivu. Poikkeava tekijä näiden edellä mainittujen välillä on käyttötarkoitus. Intranetistä löytyvä tieto on merkittävää yrityksen työntekijöille tai tässä projektissa Laurean opiskelijoille ja henkilökunnalle. Kyse on siis tiedosta jota ei ole tarkoitettu julkiseksi tai jolla on merkitystä vain yrityksen sisäisessä toiminnassa.

Yritykseen intranetiin voidaan määritellä ns. suljettuja ja avoimia osia. Avoimeen osioon ei tarvita sisäänkirjautumista, vaan tunnistautuminen tapahtuu jo käyttäjän avatessa henkilökohtaisen tietokoneensa ja kirjautuessa verkkoon. (Ek & Clason 1999, 1-3.) Suljettuihin osioihin pääsy vaatii erillisen kirjautumisen intranetissä ja se voidaan määritellä vain tietyille työntekijöille tai tietyille ryhmälle.

Intranetin avulla saavutetaan monia etuja ja sen käyttöönotosta seuraa kustannussäästöjä. Intranetin avulla sisäisen ja ulkoisen tiedon saatavuus paranee. Paperin käyttö vähenee, koska tieto kulkee pääasiassa digitaalisessa muodossa. Joustavuus lisääntyy, kun käyttäjät voivat tarkastella organisaation tuottamien asiakirjojen sisältöä oman selaimensa avulla. Tarvitaan vain tiedon sijaintipaikka ja tarvittavat oikeudet. Hyvin rakennettu intranet saa käyttäjän tuntemaan, että uusi järjestelmä on tuonut jotain uutta ja mielenkiintoista. Esimerkiksi se säästää aikaa tietyn toiminnon tekemisessä. (Ek & Clason 1999, 4-5.)

Käyttöliittymä pitää olla helppokäyttöinen ja houkutteleva, jos palvelun pääsy, käyttö tai toiminnot ovat monen mutkan takana, syntyy tyytymättömyyttä ja käyttäjät alkavat kaivata vanhaa järjestelmää. Mitä enemmän intranet sisältää johdonmukaisia, loogisia ja määriteltyjä toimintoja, sitä helpompaa sen käyttö on. Käyttäjäystävällisyys lisääntyy, jos intranetin käyttöliittymäksi valitaan ratkaisu, jonka työntekijät tuntevat entuudestaan esim. Windows-tyyppinen käyttöliittymä. Tutun käyttöliittymän myötä tarvitaan vähemmän käyttäjäkoulutusta ja intranetin opettelu helpottuu. (Ek & Clason 1999, 4, 89.)

4 Vaatimusmäärittely

Vaatimusmäärittelyn tärkeimpiä tehtäviä on toimia pohjana järjestelmän kehittämiselle ja kuvata, mitä vaatimuksia järjestelmän tulee täyttää. Määrittelyvaihe voidaan jakaa kahteen eri osaan: asiakkaan vaatimusten kartoittamiseen ja toteutettavan järjestelmän määrittelyyn. Asiakkaan vaatimuksia kartoittaessa järjestelmälle asetetaan yleisiä tavoitteita, esimerkiksi kuinka saadaan kaikki asiakirjat ja yleiset tiedot samaan paikkaan. Steve McConnell (1998, 116) jakaa vaatimusmäärittelyn kolmeen eri osaan: vaatimusehdotusten keräämiseen, vaatimusten määrittelyyn ja niiden analysointiin.

Vaatimusehdotusten kerääminen kannattaa aloittaa haastattelemalla avain- ja loppukäyttäjiä, tarkastelemalla kilpailevia tuotteita tai rakentamalla vuorovaikuttavia prototyyppejä. Vaatimusmäärittelyn tulokset kootaan kirjalliseen muotoon, joka voi olla perinteisen vaatimusmäärittelydokumentin sijasta kuvattuna tapahtumakaaviona tai vuorovaikutteisena käyttöliittymäprototyyppinä. Vaatimusten analysoinnin avulla tulevasta järjestelmästä etsitään yhteneväisyyksiä ja eroavaisuuksia kerättyjen vaatimusten väliltä ja yhdistetään vaatimukset niin, että jäljelle jää vain olennaiset piirteet. (McConnell 1998, 116-117.)

Vaatimusmäärittelyä työstettäessä valitaan tarvittava määrä joukko avain- ja loppukäyttäjiä, joiden tehtävänä on määrittellä, millaisen järjestelmän he haluavat rakennettavan. Haastatteleamalla loppukäyttäjiä selviävät järjestelmän alustavat vaatimukset. Yksinkertaisen prototyypin avulla on helppo esitellä järjestelmää ja saada myös palautetta avain- ja loppukäyttäjiltä jatkokehitystä varten. Käyttöliittymän ulkonäön ja käyttötuntuman rakentaminen onnistuu tyylioppaiden avulla, jotka kannattaa esitellä avain- ja loppukäyttäjryhmälle. Prototyyppi kannattaa laajentaa niin, että se kattaa järjestelmän kaikki toiminnalliset osa-alueet, näin ollen sitä voidaan käyttää pohjamäärittelynä tulevalle järjestelmälle. (McConnell 1998, 116-117.) Tässä projektissa vaatimusmäärittely kuvaa sivuston toiminnallisuutta ja toimi työohjeena projektin toteutusvaiheelle.

Vaatimusmäärittelyllä on monta tarkoitusta, esimerkiksi toimia vaatimukset kuvaavana dokumenttina, vaatimusten syntymiseen johtavana prosessina, yhteisenä ymmärryksenä kehittämisen kohteesta sekä yhteisenä sopimuksena vaatimuksista. Vaatimusmäärittelyn tulee perustua ymmärrykseen yrityksen liiketoiminnasta, prosesseista ja tulevista käyttäjistä. Ohjelmistoprojekteissa ei tulisi arvioida pelkkää vaatimusmäärittelyä, sen sijaan huomiota tulisi kiinnittää itse prosessiin, joka tuottaa oppimista, yhteistä ymmärrystä ja sitoutumista. Jokaisessa vaatimusmäärittelyssä on virheitä, epäselvyyksiä ja muita erovaihteluita. Vaatimusten jäsentämisessä kannattaa käyttää useampaa näkökulmaa, hyvä määrittelytyö tarkastelee asioita monesta eri näkökulmasta, eikä pelkästään esimerkiksi käyttöliittymätapausten kautta. Välttyäksemme turhilta virheiltilta ja ylimääräisiltä työtunneilta, on tärkeää että määrittely käydään perusteellisesti läpi molempien osapuolten, asiakkaan ja toimittajan kanssa. Vasta sen jälkeen voidaan alkaa parantamaan järjestelmän sisältöä. (Vuori 2009, 20-21.)

Jokaisella projektiin osallistuvalla pitäisi olla sama käsitys suunniteltavasta järjestelmästä ja sen toiminnasta. Jos ajatus konseptista ei ole selkeänä kaikkien mielessä, kehitetään asiaa, jota kukaan ei ymmärrä ja jokainen osapuoli näkee asian eri tavalla. Suunniteltavasta järjestelmä ei kannata heti tehdä suurta, vaan se kannattaa toteuttaa vaiheittain pienemmissä osissa. Asiakkaalta saatuja vaatimuksia tulee kehittää monella tapaa vaatimusmäärittelyä tuottaessa ja tämänkin jälkeen vaatimuksia tulee arvioida, täydentää ja laajentaa koko ohjelmistoprojektin ajan. Myös kokonaan uudet näkökulmat vaatimuksiin ovat huomionarvoisia eikä niitä kannata vähätellä. Vasta tämän jälkeen voidaan luottaa siihen, että määrittely on toteutettu perustellusti. (Vuori 2009, 20.)

Käyttäjätutkimuksien tekeminen on edellytys toimivalle järjestelmälle. Käyttäjätutkimuksen tekemättä jättäminen johtaa siihen, että toimittajalla ei ole kunnollista kuvaa siitä kenelle, mihin tarpeisiin ja millaiseen ympäristöön järjestelmää ollaan suunnittelemassa. Jos edellä mainitut asiat eivät ole halussa, järjestelmä ei palvele loppukäyttäjiä vaan hankaloittaa työntekoa, vaikka järjestelmän oli tarkoitus nopeuttaa ja helpottaa omaa työtä. Prototyyppien

valmistaminen helpottaa asiakkaan ymmärrystä järjestelmästä ja lisäksi sen avulla pystytään hiomaan järjestelmää entistä paremmaksi käyttäjän kannalta. Prototyyppi helpottaa toki myös toimittajan työtä, sillä silloin nähdään, mikä järjestelmässä toimii ja mikä ei. (Vuori 2009, 20-21.)

Vaatimusten hallintatyökalut eivät ole välttämättä paras vaihtoehto vaatimusten luomisvaiheessa, tässäkin ongelmaksi voi muodostua se, että hallintatyökalu piilottaa tärkeitä asioita kokonaisuudesta. Vaatimusten hallintatyökalut ovat korvaamaton apu, kun päästään vaatimusten ja muutoksen hallintaan. Vaatimusmäärittelydokumentaation tuottaminen on projektin tärkeimpiä vaiheita, joten osaavien ammattilaisten käsissä vältetään sudenkuopat ja osataan yhdistää eri osapuolten tarpeet yhtenäiseksi kokonaisuudeksi. Usein organisaatiossa jokainen kohderyhmä haluaisi kehittää järjestelmää omiin tarpeisiinsa, vaikka järjestelmän tarkoitus on palvella esim. myyntiorganisaatiota. Tätäkin asiaa pystytään hallitsemaan osavalla vaatimusmäärittelyn ja konseptoinnin prosesseilla. Ei-toiminnallisten vaatimuksien, esim. tietoturvan ja käytettävyyden, pitää olla määriteltynä, jotta tuleva järjestelmä pystytään testaamaan kunnolla. (Vuori 2009, 20-21.)

Priorisoinnin avulla varmistetaan tulevan järjestelmän toimivuus ja käytettävyys, sen takia on tehtävä päätöksiä siitä, mitkä käyttötarkoitukset ja kohderyhmät ovat tärkeämpiä kuin toiset. Keskeneräisessä järjestelmässä kaikki käyttötarkoitukset ja kohderyhmät koetaan yhtä tärkeiksi. Riskianalyysien tekemiseen on panostettava jo projektin alkuvaiheessa, jotta niistä on hyötyä tulevassa projektissa. Riskianalyysien avulla opitaan tunnistamaan järjestelmän kriittisiä piirteitä. Hyvä vaatimusmäärittely rajaa vain pakolliset reunaehdot, mutta jättää tilaa myös suunnittelulle, joten sen toteuttamista ei kannata antaa kenen tahansa tehtäväksi. Kyseessä on projektin kriittisin vaihe ja osaava ammattilainen tietää, miten välttää sudenkuopat ja yhdistää eri käyttäjien tarpeet yhtenäiseksi kokonaisuudeksi. (Vuori 2009, 20-21.)

On selvää, että edellä mainittuja asioita esiintyy eritasoisina ja eri laajuudessa riippuen siitä, onko vaatimusmäärittelyn tekijänä toimittaja, asiakas vai kolmas osapuoli. Tunnistamalla mahdollisia riskejä omassa projektissa voidaan välttää järjestelmähankkeen vaarantuminen. (Vuori 2009, 21.)

5 Vaatimusten hallinta

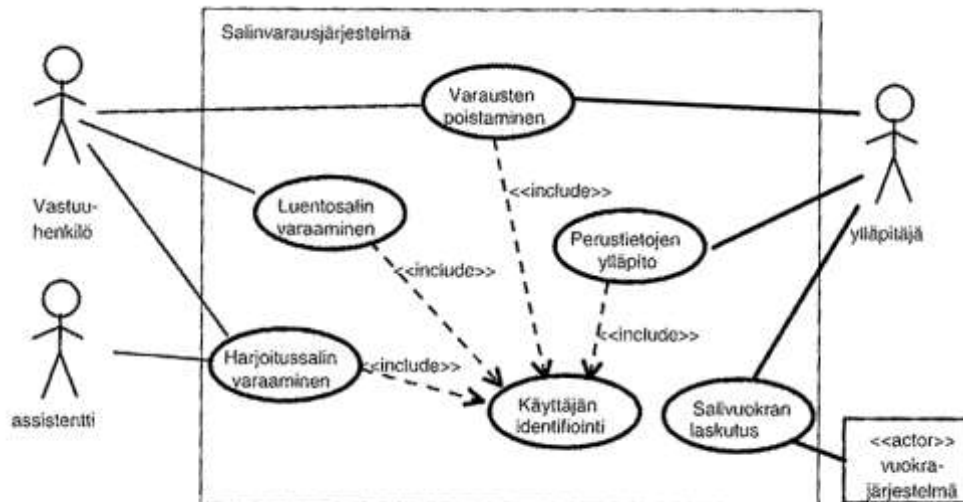
Vaatimusten hallinta käsitteenä liitetään useimmiten ohjelmistotuotantoon. Puhuttaessa vaatimustenhallinnasta sillä tarkoitetaan yleensä järjestelmällistä varmistusta siitä, että valmis tietojärjestelmä sisältää kaikki asiakkaan haluamat ominaisuudet - toisin sanoen asiakkaan esittämät vaatimukset. Tärkeänä osana vaatimusten hallintaa pidetään muutosten hallintaa ja jäljitettävyyttä. Vaatimusten hallintaa ei pidetä erillisenä tukitoimintona ohjelmistoprojek-

teissa, vaan se sisällytetään projektin muihin osa-alueisiin, kuten esitutkimukseen, vaatimusmäärittelyyn sekä tuotteen- ja projektinhallintaan. (Haikala & Märijärvi 2004, 91-92.)

Usein ohjelmistoprojekteja toteutettaessa vaatimusten hallintaan ei panosteta riittävästi. Tästä seuraa, että se, mitä oli ajateltu tehtävän, ei toteutunut ja lopputulos on aivan jotain muuta. Vaatimusten hallinnassa tärkeää on oikein ymmärretyt asiakasvaatimukset. Asiakasvaatimukset ja niistä työstettävät ohjelmistovaatimukset selviävät projektin alkuvaiheessa esitutkimus- ja määrittelyvaiheessa. Asiakasvaatimukset saadaan mm. markkinoinnilta, omasta organisaatiosta, käyttäjiltä tai aivoriihen tuloksena. Määritellyt asiakasvaatimukset muuttuvat projektin aikana, tämä saattaa johtua käyttäjien erilaisista tai ristiriitaisista tarpeista. Muuttuva markkinatilanne voi vaikuttaa järjestelmän sisältämiin ominaisuuksiin. Yrityksen liiketoiminnallisia asioita mm. kustannus-, hyöty- ja markkina-analyysijä liitetään vaatimustenhallintaan, jotta varmistetaan tulevan järjestelmän todelliset ominaisuudet. Viimeistään muutoksia järjestelmään tuo sen ylläpito tai uusien versioiden luominen. Asiakkaan tekemät muutokset järjestelmään vaikuttavat ohjelmistotyön kaikkiin vaiheisiin. Selvää on, että mitä myöhemmässä vaiheessa muutoksia tehdään, sitä enemmän lisätyötä ne aiheuttavat. Vaatimusten hallintaa tulisi tehdä koko järjestelmän elinkaaren ajan, eikä pelkästään esitutkimus- ja määrittelyvaiheessa. Vaatimusten muutosten hallinta on yksi tärkeimmistä asioista vaatimusten hallinnassa, joten on tärkeää sopia miten mahdollisista muutoksista sovitaan. (Haikala & Märijärvi 2004, 92-99.)

Käyttötapausten tarkoitus on toimia kommunikointivälineenä asiakasvaatimuksia kartoitettaessa ja kuvattaessa niitä ohjelmistovaatimuksiksi. Käyttötapaukset eivät korvaa toiminnallissa määrittelyssä esiteltäviä järjestelmän toimintojen kuvauksia. Käyttötapauksia kannattaa hyödyntää ohjelmistoprojektin muissa vaiheissa. Käyttötapaukset luovat hyvän pohjan järjestelmätestaukselle ja käyttö-ohjeiden kirjoittamiselle. Käyttötapauksia ei voida hyödyntää arkkitehtuurisuunnittelussa, koska käyttötapaus kuvaa kahden tai useamman moduulin välistä yhteistyötä. (Haikala & Märijärvi 2004, 160-161.)

Asiakasvaatimuksia ja niiden liittymistä vaatimusmäärittelyyn voidaan kuvata ns. käyttötapausten (use case) avulla (Kuva 1). Käyttötapausten ideana on kuvata järjestelmän toiminnallisuus käyttäjien sille suorittamina tapahtumaketjuina. Järjestelmän käyttötapauksia kuvataan piirroksilla ja kuhunkin käyttötapaukseen liittyy yksi tai useampia käyttäjärooleja (actor), joka suorittaa käyttötapausten. Järjestelmän käyttötapaukset ja käyttäjäroolit kuvataan ns. käyttötapauskaaviossa (use-case diagram). (Haikala & Märijärvi 2004, 158-159.)



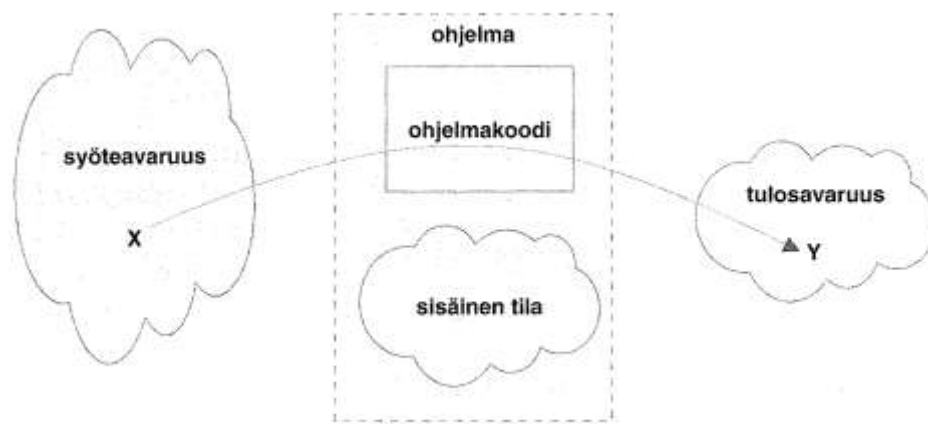
Kuva 1: Käyttötapauskaavio (Haikala & Märijärvi 2004, 158).

Käyttötapaus alkaa aina käyttäjäroolin aloitteesta ja päättyy järjestelmän tuottamaan lisäarvoon eli käyttäjä on saanut suoritettua mielekkään tehtävän tai tehtäväkokonaisuuden. Esimerkiksi hissien tilaaminen ei ole vielä hyvä käyttötapaus, sillä hissien saapumisella ei ole merkitystä järjestelmän tarjoaman palvelun kannalta. Käyttötapaukseen tulee sisältyä matkustaminen hississä toiseen kerrokseen. Hyvällä käyttötapauksella on seuraavia ominaisuuksia: Asiakkaan ja järjestelmän tulevan käyttäjän on ymmärrettävä käyttötapaukset, joten niiden tulee olla todellisia. Asiakasvaatimukset tulee kuvata, ottamatta kantaa järjestelmän toteutukseen, esim. ”käyttäjä kirjautuu sisään” on parempi ilmaus kuin ”käyttäjä kirjautuu sisään syöttämällä kuusinumeroisen opiskelijatunnuksen”. Käyttötapausten on oltava testattavia, sillä ne muodostavat perustan järjestelmätestaukselle. Käyttötapausten on muodostettava kokonaisuus, jotta se voidaan testata yhtenä tai useampana peräkkäisenä testitapauksena. Sopivan kokoinen käyttötapaus mahtuu yhdelle A4-kokoiselle paperille, muuten käyttötapaus on liian laaja. Käyttötapauksiin ei voi ottaa mukaan kaikkia yksityiskohtia, vaan niiden avulla kuvataan tärkeimmät osat tulevasta järjestelmästä. (Haikala & Märijärvi 2004, 159.)

6 Testaus

Testaus on ohjelman suorittamista virheiden löytämiseksi. Haikala & Märijärvi (2004) määrittelevät järjestelmien tai ohjelmistojen testauksen suunnitelmalliseksi virheiden etsimiseksi ohjelmaa tai sen osaa suorittamalla. Testaus tapahtuu usein kokeilemalla ohjelmaa ennalta määrittelemättömillä syötteillä. Tällaisessa tapauksessa ei ole tarkoitus etsiä virheitä ohjelmakoodista, vaan osoittaa ohjelman toimivuus. Testaukseen liittyvät työvaiheet ovat testauksen suunnittelu (testaussuunnitelma, testitapaukset), testausympäristön luominen, testauksen suorittaminen ja tulosten analysointi. (Haikala & Märijärvi 2004, 283-284.)

Testaustulos riippuu paitsi syötteestä, myös järjestelmän sisäisestä tilasta. Sisäisellä tilalla tarkoitetaan ohjelman muuttujien arvoja, rekisterien arvoja tai levyllä tallennettuja tietoja. Testin onnistuminen edellyttää että yksittäisen testin tulos on odotetun kaltainen (Kuva 2). Tämä edellyttää sitä, että järjestelmästä on olemassa kattava vaatimusmäärittely. Kattavan vaatimusmäärittelyn perusteella voidaan suunnitella testitapaukset ja todeta hyväksyttävät lopputulokset. Ohjelmakoodi monimutkaisuudessaan sisältää virheitä. Testauksen tavoitteena on minimoida virheitä. Ohjelman virheettömyyttä testauksella ei pystytä osoittamaan, edes yksinkertaisissa tapauksissa. Haikala ja Märijärvi (2004, 285-287) korostavat ohjelmakoodin huolellista suunnittelua ja ohjelmoinnin merkitystä.

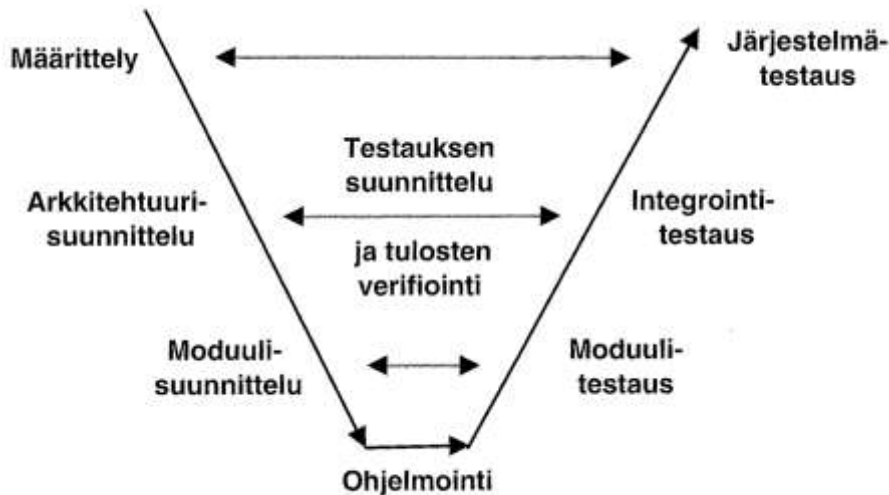


Kuva 2: Ohjelmiston testaus (Haikala & Märijärvi 2004, 284).

Virhe on poikkeama ohjelmiston toiminnassa ja sille tuotetuissa määrittelyissä. Virheet voidaan jakaa kääntäjän käännöksen aikaisiin virheisiin, ajon aikaisiin ja loogisiin virheisiin. Testauksen vaihemallin mukaan virheet voidaan jakaa virheellisten vaatimusten määrittelyyn, suunnittelu- ja ohjelmointivirheisiin. (Räsänen 2002.) Virheen vakavuus voi vaihdella järjestelmän käytön kokonaan estävästä virheestä käyttäjää ärsyttävään yksityiskohtaan (Haikala & Märijärvi 2004, 287).

V-malli (Kuva 3) on prosessimalli, jota voidaan käyttää ohjelmistoprojektin suunnittelussa ja toteutuksessa. V-mallin avulla on helppo määritellä, mitä kukin tekee ja missä vaiheessa projektia. V-malli voidaan muokata sopivaksi jokaista projektia ja organisaatiota varten. V-malli lähtee liikkeelle suunnittelusta, joka löytyy V:n vasemman sakaran yläreunasta, josta lähdetään etenemään alaspäin. V-mallissa yläosassa on vaatimusmäärittely, keskiosassa on suunnitteluvaihe ja V:n pohjalla tuotteen toteutus. Viimeisessä vaiheessa V-mallissa on testaus, jolloin lähdetään V:n sakaraan ylöspäin oikealle. Testausvaiheessa verrataan koko ajan että vastaako järjestelmä samalla tasolla olevaa suunnitteluvaihetta. Ensimmäisenä testataan tuot-

teet toiminnot, sitten koko tuotteen toiminta ja lopuksi testi järjestelmän loppukäyttäjillä. (Wikipedia 2010.)



Kuva 3: Testauksen V-malli (Haikala & Märijärvi 2004, 289).

Moduulitestauksessa testattavana on yksittäinen moduuli. Testattava moduuli koostuu yleensä noin 100-1000 ohjelmarivistä. Sen toimintaa verrataan tekniseen määrittelydokumenttiin. (Haikala & Märijärvi 2004, 289.)

Integrointitestauksessa siirrytään askel eteenpäin ja yhdistetään moduuleita keskenään. Integrointitestauksessa ei enää keskitytä tarkastelemaan yksittäistä moduulia, vaan moduulien välisten rajapintojen toimivuutta. Kuten moduulitestauksessa integrointitestauksessakin tuloksia verrataan tekniseen dokumentaatioon. Yleensä integrointitestausta suoritetaan rinnakkain moduulitestauksen kanssa. (Haikala & Märijärvi 2004, 290.)

Järjestelmätestauksessa testataan koko järjestelmää ja tuloksia verrataan määrittelydokumentaatioon ja asiakasvaatimuksiin. Järjestelmätestaukseen voidaan liittää kenttätестaus ja hyväksymistestaus (Haikala & Märijärvi 2004, 288-291). Kenttätестaukset toteutetaan esim. testauspäällikön valvonnassa, järjestelmän lopullisessa käyttöympäristössä. Hyväksymistestaus toteutetaan järjestelmän lopullisessa käyttöympäristössä asiakkaan toteuttamana (Tauriainen 2005). Järjestelmätestauksen suorittajien tulisi olla loppukäyttäjiä, jotka eivät ole mukana ohjelmiston kehitystyössä. Testaus voidaan toteuttaa asiakkaan tai toimittajan tiloissa. Alfa-testauksessa asiakas toteuttaa testauksen toimittajan tiloissa. Vastaavasti beta-testauksessa asiakas testaa itse järjestelmää omilla tiloilla. (Haikala & Märijärvi 2004, 288-291.)

Jossakin tapauksissa järjestelmä joudutaan testaamaan uudelleen, jota kutsutaan regressiotestaukseksi. Kun järjestelmätestauksessa havaitaan virhe, saattaa korjaus aiheuttaa muutoksia useisiin moduuleihin. Jotta kaikki mahdolliset muutostarpeet tulisi huomioitua, saatetaan testaus joutua uusimaan muille moduuleille. Mitä ylemmäksi V-mallissa edetään, sitä kalliimpaa testaaminen on, tästä syystä on tärkeää panostaa moduuli- ja integraatiotestaukseen, jotta vältetään uudelleentestaus järjestelmätasolla. (Haikala & Märijärvi 2004, 290.)

Järjestelmää voidaan testata käyttöliittymän osalta. Käytettävyytestauksella pyritään selvittämään, toimiiko rakenteilla oleva järjestelmä niin, että käyttäjä pystyy suoriutumaan tehtävistä, joita järjestelmältä vaaditaan. Käytettävyytestauksessa keskitytään testaamaan pelkästään käyttöliittymää. Käyttöliittymätestausta tehdään jo usein määrittelyvaiheessa käyttöliittymäprototyypin avulla. Käytettävyytestaus voidaan suorittaa esim. käytettävyyslaboratoriossa. Videoinnin avulla voidaan tallettaa testaajan ajatukset siitä miten hän käyttää järjestelmää. Käytettävyys testauksessa voidaan hyödyntää ohjelmistojen käytettävyyteen perehtynyttä asiantuntijaa. (Haikala & Märijärvi 2004, 291.)

Testitapausten valintaan voidaan käyttää kahta erilaista tapaa: lasilaatikkotestausta tai mustalaatikkotestausta. Suurin ero näiden kahden tavan välillä on se, että mustalaatikkotestaus perustuu yksinomaan testattavan järjestelmän dokumentaatioon tutustumatta itse järjestelmään. Lasilaatikkotestauksessa hyödynnetään vastaavasti järjestelmän toteutusta eli ohjelmakoodia. Harmaalaatikkotestauksessa hyödynnetään järjestelmän toteutusperiaatteita eli miten yksittäinen ohjelman osio toimii. (Haikala & Märijärvi 2004, 291).

Testauksen dokumentointi voi sisältää useita dokumentteja, pienemmissä projekteissa on tarpeen tehdä vain yksi testaussuunnitelma. Testaussuunnitelmasta pitäisi käydä ilmi ainakin, mitä testejä tehdään ja milloin, miten ne järjestetään ja millaisia lopputuloksia odotetaan. Testausdokumentaatioon on olennaista merkitä testauksen lopettamiskriteerit. Kaikki testauksen aikana löytyneet virheet tulisi raportoida ja analysoida riippumatta testauksen tasosta. Lisäksi kirjattavia tietoja ovat mm. virheen kuvaus, miten vakavasta virheestä on kysymys, milloin virhe löydettiin. (Haikala & Märijärvi 2004, 299-300.)

7 Laurean Intranetin testaus

Testauksen suunnittelussa hyödynsin järjestelmästä laadittua vaatimusmäärittelyä. Testitapausten valinnassa käytin mustalaatikkotestausta, koska minulla ei ollut mahdollisuutta tutustua itse järjestelmään tai sen toimintoihin, vaan testikohteet piti suunnitella vaatimusmäärittelyn pohjalta. Testauslomakkeet oli määrä saada valmiiksi joulukuun loppuun mennessä, sillä testaus aloitettiin tammikuun alussa. Testaukset toteutettiin lähettämällä kullekin testaajalle suunnitellut lomakkeet sähköisessä muodossa ja he omatoimisesti testasivat, täyttivät lomakkeet ja palauttivat ne testausvastaavalle. Testiin valitut kaksi opiskelijaa testasivat järjes-

telmän ryhmättestauksena Laurean tiloissa. Tämän jälkeen osa testaajista kävi läpi järjestelmää myös toimittajan laatiman testauslomakkeen pohjalta (Liite 2).

Tutkittuani testaustuloksia, huomasin, että järjestelmä on ollut hyvin keskeneräinen ja puolet toiminnoista antoi virheellisen testituloksen, joka on katastrofaalinen eli tasoa 1. Osassa testitapauksia virheellinen testitulos on johtunut siitä, että järjestelmään ei ollut ehditty syöttää tarpeeksi tietoja testausta varten.

Kehittelemämme testauslomakkeet (Liite 1) olivat kaikille testaajille samanlaiset ja sisälsivät seuraavat kentät:

- Testikohde
- Toivottu tapahtuma
- Tulos
- Virheen laatu
- Muita huomioita/Kommentteja

Testikohde kertoo testaajalle mitä asiaa tulee testata (selitetty sanallisesti). Toivottu tapahtuma puolestaan kertoo, miten järjestelmän tulisi reagoida, jos se toimii oikein. Tulos sarakkeeseen testaaja merkitsee OK, joka tarkoittaa, että tulos on hyväksytty ja järjestelmä antaa toivotun tapahtuman testattavasti kohteesta. V-merkintä tulos-sarakkeessa tarkoittaa, että testauksessa on tullut virhe ja toivottua tapahtumaa ei tullut. Lisäksi testilomakkeessa arvioitiin virheen laatua neljällä eri kriteerillä:

- 1= Katastrofaalinen virhe, esim. järjestelmä kaatuu, kun toimintoja suoritetaan
- 2=Paljon toimintaa haittaava virhe, esim. toimintoa suoritettaessa järjestelmä hidastuu huomattavasti
- 3=Kiusallinen virhe, esim. asiat täytyy tehdä tietyssä järjestyksessä, jotta osion saa toimimaan
- 4=kosmeettinen virhe, esim. hakutulokset listautuvat huonosti

Testaajat jakautuivat kahteen eri pääkäyttäjryhmään: sisällönsyöttäjiin ja loppukäyttäjiin. Sisällönsyöttäjät koostuivat pääasiassa Laurean viestinnän henkilökunnasta ja loppukäyttäjiin kuului Laurean opiskelijoita ja henkilökuntaa.

Testausta suoritettiin 15 testaajan voimin, mutta matkan varrella kaikilla testaajille ei ollut aikaa tehdä omaa osuuttaan ja näin saimme loppuen lopuksi koottua 13 testaajan joukon. Täytetyn testauslomakkeen saimme takaisin kuudelta testaajalta eli puolet näistä edellä mainitsemistani 13:sta ei palauttanut koskaan testauslomaketta. Lisäksi järjestelmälle tehtiin

englanninkielinen testaus, tälle ei toteutettu erikseen omaa lomaketta, vaan päätimme, että käytämme samoja lomakkeita kuin suomenkielisen osion testaamiseen.

Testaukset aloitettiin tammikuun toinen päivä, mutta ne jouduttiin keskeyttämään kahdeksi viikoksi järjestelmän ollessa liian keskeneräinen hyväksymistestauksen suorittamiseen. Kun toimittaja oli ilmoittanut meille, että järjestelmän toimitus tulee siirtymään, harkitsimme kuitenkin keskeneräisen version testausta opiskelijoiden kanssa. Päätimme kuitenkin luopua ajatuksesta, sillä raakileen testaamisesta ei olisi loppuen lopuksi ollut mitään merkittävää hyötyä projektin kannalta. Lisäksi testauksia viivästytti sisällön puuttuminen. Järjestelmään ei ollut ehditty syöttää tarpeeksi sisältöä.

8 Arviointi

Mielestäni projekti ei tuonut testaukseen mitään uutta ja innovatiivista. Kyseessä oli toiminnallinen opinnäytetyö, jonka tarkoituksena oli toteuttaa Laurean tilaamalle järjestelmälle hyväksymistestaus ja raportoida tulokset. Itse projektin aikana en oppinut mitään uutta, joutuin siitä, että osallistumiseni projektiin jäi vähäiseksi. Kirjoittaessani tätä opinnäytetyötä, olen oppinut ohjelmistoprojektista, testauksesta ja vaatimusmäärittelystä enemmän kuin koko varsinaisen projektin aikana. Olisin halunnut perehtyä enemmän testauksen ja testauslomakkeiden suunnitteluun, mutta minulla ei ollut tarpeeksi aikaa. Monet asiat jäivät mietittämättä, kuten kuinka paljon laatimiani lomakkeita hyödynnettiin testauksen jälkeen havaittujen virheiden korjaamiseen. Olisi mielenkiintoista tutkia, miten tämän projektin vaatimusmäärittely onnistui: miten paljon valmis järjestelmä vastasi laadittua vaatimusmäärittelyä.

Heti projektin alusta lähtien olisin kaivannut enemmän tukea ohjaavalta opettajalta. Useasti projektin aikana minusta tuntui siltä, että en oikein tiennyt mitä minun olisi pitänyt seuravaksi tehdä. Epävarmuuttani lisäsi se, ettei minulla ollut toista opiskelijaa auttamassa projektissa. Jos työtä olisi ollut toteuttamassa useampi ihminen, olisivat tulokset voineet olla aivan toisenlaiset, sillä silloin olisi pystynyt pyytämään toisen mielipiteen epävarmassa tilanteessa. Palautteen saaminen on todella tärkeää oman motivaation ja jaksamisen kannalta, koska se antaa suuntaa myös sille, ollaanko tekemässä oikeita asioita vai hukataanko aikaa väärin asioiden tutkimiseen. Lähetin testauslomakkeet sekä testaus suunnitelman luettavaksi ohjaavalle opettajalle ja hänen mielestään ne olivat hyviä. Tässäkin tapauksessa ohjaava opettaja olisi voinut olla kriittisempi ja kehottaa miettimään testauslomakkeiden toimivuutta. Koko projektin aikana en saanut palautetta testauslomakkeiden onnistumisesta tai ylipäätään oman osuuteni hoitamisesta tässä projektissa. Parhaimmassa tapauksessa projekti olisi kasvattanut omaa asiantuntijuutta, mutta tällä kertaa kävi toisin. Asiantuntijuuttani kasvatti enemmän tämän opinnäytetyön kirjoittaminen. Mietin näin jälkikäteen, että minun olisi pitänyt vaihtaa toiseen aiheeseen, jotta olisin itse saanut paremman hyödyn opinnäytetyöstä.

Kaikesta huolimatta onnistuin mielestäni hyvin testauslomakkeiden kehittämisessä, vaikka niiden kokoaminen tuntui aika ajoin todella haastavalta, sillä en ollut aikaisemmin toteuttanut hyväksymistestausta tai edes testauslomakkeita. Toimittajan eli Onesta Solutionin testauslomake (Liite 2) oli mielestämme hyvin samantyyppinen kuin kehittämämme testauslomake. Onestan lomakkeessa testattavia osioita oli yhteensä 13. Yhdessä osiossa oli paljon enemmän testitapauksia kuin kehittämässämme testauslomakkeissa.

Jälkeenpäin huomasimme, että emme olleet osanneet ottaa huomioon kaikkia yksityiskohtia suunnitellessamme ja toteuttaessamme omia testauslomakkeita. Saimme palautetta muun muassa siitä, että testauslomakkeissa olisi pitänyt olla selkeämmät ohjeet mitä tulee tehdä. Lomakkeet olisi kannattanut käydä läpi yhdessä testaaajien kanssa ennen hyväksymistestausta. Testaaajilta olisimme saaneet varmasti arvokasta tietoa ja mahdollisia parannusehdotuksia testauslomakkeita koskien. Edellä mainituilla toimenpiteillä testauksesta olisi saatu parempia tuloksia ja enemmän palautuneita lomakkeita.

Toinen vaihtoehto olisi ollut, että olisimme ohjanneet testaaajia henkilökohtaisesti testaustilanteessa lomakkeen täyttämässä ja selventäneet tarvittaessa epäselviä kohtia. Toisaalta henkilökohtainen ohjaus ei olisi onnistunut, sillä testaaajat suorittivat testaukset omalla koneella sekä omalla ajalla. Toki olimme tavoitettavissa sähköpostilla ja puhelimitse, mutta tuki ei ole samanlaista kuin henkilökohtaisessa ohjauksessa. Kaikki testaaajat eivät kokeneet saaneensa tarpeeksi apua testaamisen suorittamiseksi, vaan joutuivat päättämään itse testauslomakkeiden perusteella, mitä asioita tulee testata.

Arvioidessamme jälkeenpäin testaustuloksia huomasimme, että joissakin lomakkeissa testattiin turhaan samaa asiaa moneen kertaan erotuksena vain erilaiset syötteet. Testikohteet olivat kuvattu sanallisesti vaatimusmäärittelyn mukaisesti. Testikohteiden kuvaaminen käytötapausten kautta olisi toiminut mielestäni paremmin.

Tarkoituksena oli käydä saamamme testauslomakkeet läpi projektiassistentti Heidi Ahon kanssa, joka toimi yhtenä testaaajista. Kun testitulosten läpikäynnin piti alkaa, saimme ilmoituksen, ettei Heidi enää ollut Laurean palveluksessa. Loppuen lopuksi testitulosten läpikäynti jäi kokonaan minulle, eikä Heidin tilalle hankittu ketään muuta projektiryhmästä.

Projektin aikana huomasin, että en painottanut tarpeeksi aikataulua lomakkeiden palautuksen suhteen. Testauslomakkeet palautettiin aikataulusta myöhässä, vaikka olimme Merja Jalavan kanssa laatineet yhteenvedon testausaikataulusta (Liite 3). Sama testausaikataulu jaettiin sähköisesti kaikille testaajille. Toisaalta, kuten aikaisemmin jo mainitsin, halusimme helpot-

taa testaajien työtä, jolloin he saivat omien aikataulujensa puitteissa toteuttaa testaukset, tämäkin saattoi olla syynä palautusaikataulun venymiseen.

Projektin aikana tein havainnon, kuinka tärkeää on olla mukana jo alusta saakka projektissa, jotta pystyy mahdollisemman hyvin hahmottamaan järjestelmän rakenteen ja ulkoasun. Suurimmaksi puutteeksi projektin aikana koin, etten päässyt osallistumaan projektipalaveriiniin. Kuten jo alussa mainitsin, käsitys järjestelmästä ja sen toiminnasta jäi todella puutteelliseksi, sillä kokonaiskuva järjestelmästä piti muodostaa pelkän vaatimusmäärittelyn ja muutaman käyttöliittymämallin pohjalta.

Kirjoittaessani tätä työtä olen alkanut hahmottaa, miten paljon erilaisia asioita on otettava huomioon järjestelmän testauksessa. Toki siihen liittyy vielä suunnitteluvaihe, koodaus, laadunvarmistus, mutta jo itse testaus on todella iso ja tärkeä osa ohjelmistoprojektia. Testaamista saatetaan vähätellä ja pitää toisarvoisena ajatellen koko ohjelmistoprojektia, mutta on todella tärkeää suunnitella ja toteuttaa testaukset mahdollisimman hyvin, jotta välttyään ylimääräisiltä kustannuksilta.

Kehitysehdotuksena tulee mieleen useamman eri toimittajan kilpailuttaminen, ennekuin ryntätään suoraan valitsemaan ensimmäinen mahdollinen järjestelmätoimittaja. Näin isossa projektissa mukana olevien henkilöiden pitää olla motivoituneita ja sitoutuneita hoitamaan projekti alusta loppuun. Ainoa onnistumisena koin testauslomakkeiden valmiiksi saamisen määritellyn aikataulun puitteissa.

9 Laurea Intranet tänään

Haastattelin puhelimitse Laurean viestinnässä työskentelevää Sanna Vanhataloa. Sanna oli myös mukana samaisessa projektissa kanssani. Hän kertoi muutamia ajatuksiaan nykyisen sivuston ongelmista ja kertoi jo hieman tulevasta intranet-projektista, joka tullaan aloittamaan joulukuussa 2010. Tuleva projekti käsittää Laurean ulkoisen- ja sisäisen verkon suunnittelun ja toteutuksen. Tällä hetkellä Laureassa on jo käytössä uudella tekniikalla toteutettu ulkoinen verkko eli Internet-sivusto.

Nykyinen järjestelmä on rakennettu vääränlaiselle ja liian jäykälle käyttöalustalle. Liian jäykkä käyttöalusta estää järjestelmän rakenteen muuttamisen, joten pientenkin rakenteellisten muutosten tekeminen on monen työtunnin koitos. Laurea Intranet sisältää paljon tietoa ja erilaisia dokumentteja. Lisähaasteita tuottaa se, että Laureassa työskentelee paljon sisällöntuottajia. Samaan aikaan sisältöä voi syöttää jopa sata ihmistä yhtä aikaa. Sisällöntuottajat päivittävät ja lisäävät tietoa Laurean Internetiin ja intranetiin päivittäin oman työnsä ohessa,

tämä aiheuttaa nykyisessä intranetissä sen että kun useampi ihminen päivittää samoja asioita järjestelmä ei pysty hallitsemaan näin valtavaa tietomäärää. (Vanhatalo 2010.)

Myöskään hakukone toimi ollenkaan nykyisessä järjestelmässä. Hakukoneen on oltava kattava ja toimiva, jotta järjestelmän käyttäjät löytävät haluamansa tiedon laajasta tiedon ja dokumenttien kirjosta. Tällä hetkellä intranetiin pystyy lisäämään ainoastaan multimedia-tiedostoja, mikä ei vastaa ollenkaan sitä kuvaa, mikä järjestelmästä oli alun perin annettu. Järjestelmään olisi pitänyt pystyä liittämään vaivatta kuvia, grafiikkaa yms., mutta näin ei kuitenkaan ole. Lisäksi käyttäjillä pitäisi olla oikeudet päästä muokkaamaan ainakin omaa profiiliaan ja tekemään rajoituksia esim. tiedotteista ym. Myös tämä järjestelmän osio toimii satumanvaraisesti, jos ollenkaan. Näin ollen järjestelmässä on edelleen vakavia puutteita, jotka aiheuttavat käyttäjille lisätyötä ja vievät työaikaa. Haastateltavan mukaan koko projekti oli aikamoinen katastrofi, joten siitä on otettu opiksi ja tulevassa projektissa asiat on tehty toisella tavalla. (Vanhatalo 2010.)

Tällä hetkellä käytössä olevan Intranet-sivuston tilalle ollaan hankkimassa uutta järjestelmää. Uuden järjestelmän vaatimusmäärittelyssä on käytetty hyödyksi Laurea Intranet-projektia varten toteutettua vaatimusmäärittelyä. (Vanhatalo 2010.) Hankkeen tavoitteena on ottaa käyttöön viestinnällisten verkkopalveluiden uusi tekninen alusta sekä uudistaa tälle alustalle Laurean pääsivusto, yksiköiden kotisivut ja Laurean Intranet.

Hankkeessa on tarkoitus toteuttaa mahdollisimman esteetön verkkopalvelu, joka vastaa Laurean verkkopalveluiden kaikkien käyttäjäryhmien tarpeita. Palvelun tulee sisältää monipuolisesti vuorovaikutteisia palveluja, joiden tavoitteena on helpottaa käyttäjien tiedonsaantia ja kommunikointia sekä edistää käyttäjien välistä yhteistyötä Laurean toiminta-alueella. (Korhonen, Pettinen & Vanhatalo 2010, 5.)

Tulevassa verkkopalvelussa on haluttu kiinnittää huomiota ennen kaikkea palvelun käytettävyyteen, riippumatta siitä onko kyseessä loppukäyttäjä vai sisällöntuottaja. Verkkopalvelun käyttämisen tulee olla helppoa ja innostavaa, riippumatta siitä mikä käyttäjän tekninen tausta tai aikaisempi verkonkäyttöhistoria on. Verkkopalvelun navigointi ja tiedon ryhmittely perustuu käyttäjälähtöisyyteen ja lisäksi tiedon hakeminen sisältyy tärkeänä osana käytettävyyteen. Tästä syystä verkkopalveluun toteutetaan tehokas hakukone ja tieto luokitellaan niin, että jokainen käyttäjä pystyy hakemaan vain hänelle oleellista tietoa. (Korhonen ym. 2010, 6-7.)

Tulevan järjestelmän rakenne tulee olemaan helposti laajennettavissa ja integroitavissa muihin järjestelmiin sekä sitä tulee pystyä kehittämään jatkossa helposti. Selaimena käytetään Microsoft Internet Explorer 8:aa. Tuleva verkkopalvelu tullaan toteuttamaan IE 8:n pohjalle,

mutta sen tulee toimia myös muilla selaimilla, kuten Firefox:lla tai Operalla. (Korhonen ym. 2010, 31.)

Selvästikin Laureassa on otettu opikseen edellisestä projektista. Lukiessani tulevan järjestelmän vaatimusmäärittelyä tunsin, että nyt ollaan menossa oikeaan suuntaan. Toimittajien kilpailutuksen ja käyttäjälähtöisen suunnittelun avulla saadaan varmasti aikaiseksi entistäkin parempi intranet Laurealle.

Lähteet

Ek, J. & Clason, M. 1999. Opi helposti intranet. Risto Torkkeli. Vantaa: Tummavuoren kirjapaino.

Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. 10., uudistettu painos. Hämeenlinna: Karisto.

Korhonen, P., Pettinen, K. & Vanhatalo, S. 2010. Vaatimusmäärittelydokumentti: Verkkopalvelualustan tekninen uudistaminen. Laurea-ammattikorkeakoulu. Laurea Tikkurila.

McConnell, S. 1998. Ohjelmistoprojektit-selviytymisopas. Marko Juoperi. Jyväskylä: Gummerus Kirjapaino.

Räsänen, S. 2002. Ohjelmiston testaus ja laatu - Testaus yleistä. Viitattu 21.10.2010.
http://webd.savonia.fi/home/ktrasse/muut/testaus_laatu/testaus_1.pdf

Suur-Askola, K. 2007. Laurea-ammattikorkeakoulu Intranet-palvelu - Projektisuunnitelma. Helsinki: Onesta Solution.

Tauriainen, S. 2005. Ohjelmistotestauksen kehittäminen. Viitattu 21.10.2010.
https://publications.theseus.fi/bitstream/handle/10024/7818/SimoTauriainen_k05_jatkot.pdf?sequence=1

Vanhatalo, S. 2010. Haastattelu 21.10.2010. Laurea-ammattikorkeakoulu. Vantaa.

Vuori, M. 2009. Vaatimusmäärittelyn huonoimmat käytännöt. Systeemityö 9/2009, 20-21.

Wikipedia. V-malli. Viitattu 21.10.2010.
<http://fi.wikipedia.org/wiki/V-malli>

Kuvat

Kuva 1: Käyttötapauskaavio (Haikala & Märijärvi 2004, 158).	14
Kuva 2: Ohjelmiston testaus (Haikala & Märijärvi 2004, 284).	15
Kuva 3: Testauksen V-malli (Haikala & Märijärvi 2004, 289).	16

Liitteet

Liite 1. Malli Laurean testauslomakkeesta	27
Liite 2. Malli Onesta Solutionin testauslomakkeesta	28
Liite 3. Hyväksymistestauksen aikataulu	29
Liite 4. Adagen käyttöliittymämallit	30

Liite 1. Malli Laurean testauslomakkeesta

Materiaalivarasto					
	Testi kohde	Toivottu tapahtuma	Tulos	Virheen laatu	Muita huomioita/Kommentteja
1.	Kokeile luoda materiaalivaraston kansio	Järjestelmä lisää kansion hierarkiassa ylimmän kansion yhteyteen.			
	Lisää kansioon muutama tiedosto	Järjestelmä liittää luodun tiedoston kyselyn kansion elementiksi			
2.					
3.	Kokeile onnistuako äsken lisäämässä tiedoston poistaminen	Järjestelmä pyytää vahvistamaan poiston ja valitut tiedostot poistetaan järjestelmästä.			
4.	Liitä kansiossa oleva dokumentti johonkin olemassa olevaan tapahtumaan	Liitetty dokumentit näkyvät tiedotteiden/ tapahtumien hallintanäkymässä.			
5.	Tarkista päivittykö liitetiedosto tapahtuman yhteyteen?	Tapahtumat/Tiedotteet sivulla pitäisi näkyä äsken lisäämäsi tiedosto.			

Virheen laatu

- 1 = Katastrofaalinen virhe, esim. järjestelmä kaatuu, kun toimintoa suoritetaan.
- 2 = Paljon toimintaa haittaava virhe, esim. toimintoa suoritettaessa järjestelmä hidastuu huomattavasti.
- 3 = Kiusallinen virhe, esim. asiat täytyy tehdä tietyssä järjestyksessä, jotta osion saa toimimaan.
- 4 = Kosmeettinen virhe, esim. hakutulokset listautuvat huonosti.

Tulos


- Ok = Hyväksytty
- V = Virhe

Liite 2. Malli Onesta Solutionin testauslomakkeesta

Laurea Intra 2007: Järjestelmätestausdokumentti

Laurea Intra 2007: Järjestelmätestausdokumentti

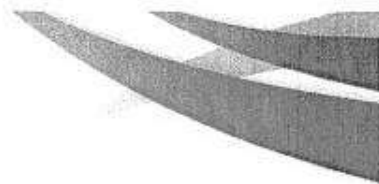
Toimipisteet

 onesta
SOLUTIONS

Tunnus	Prior.	Otsikko	Testitapaus	Rooli	Testaustulos	Toimenpiteet	OK
FUNC-6.1a	1	Laurean to	Laurea – alasuilla on vasemman palstan navigaatiassa listaus Laurean toimipisteistä ja niiden toimipistekohtaisesta sisällöstä	Sivuston käyttäjä, sisällönsyöttäjä	Ok	Ei toimenpiteitä	OK
FUNC-6.1b	1	Toimipiste	Käyttäjän on valinnut Laurean alasuilla haluamansa toimipisteen. Toimipisteen keskipalstalla käyttäjälle näytetään ko. toimipisteestä seuraavat tiedot eri alasuilla.	Sivuston käyttäjä, sisällönsyöttäjä	Ok	Ei toimenpiteitä	OK
	1		Tietoja ei näytetä taulukossa, jos sisältöä ei ko. metatiedoille ole määritelty Julkaissujärjestelmän avulla Sivuston käyttäjä voi tulostaa toimipisteen	Sivuston käyttäjä, sisällönsyöttäjä	Ok	Ei toimenpiteitä	OK
	1		"Tulostettava sivu" linkin avulla (FUNC-5.1f). Lisäksi sivulla näytetään murupolku, linkit Palaute, Sivukartta ja linkki sivun ylläpitoon.	Sivuston käyttäjä, sisällönsyöttäjä	Tulostus-painike näkyy. Ei toimenpiteitä	Ei toimenpiteitä	OK

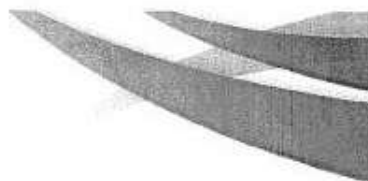
Liite 3. Hyväksymistestauksen aikataulu

INTRAN TESTAUSAIKATAULU			
Käyttäjärühmä	Testauksen ajankohta	Testattavat osiot	
Sisällönsyöttäjät			
Sari Kinnaslampi	11.-16.1	• Opinnot	
Mina Marjamaa	11.-16.1	• Kirjasto-rakenteen muutos	
Sirpa Anttila	11.-16.1	• Toimipisteet	
Marijo Valjakka	11.-16.1	• Toimipisteet	
Hanna Liinatainen	11.-16.1	• Toimipisteet	
Loppukäyttäjät			
Anna Perttö	17.-18.1	• I-asemalta haku • Kirjautuminen • Tarkennettu haku/Sisältö • Tapahtumaan ilmoittautuminen	
Eeva Haikonen	17.-18.1	• Oman profiilin päivittäminen • Toimipisteiden selaaminen ja lukeminen • Tarkennettu haku/Henkilötieto	
Kari Haimo	17.-18.1	• Etusiun selaaminen • Sisältösiivujen selaaminen ja lukeminen • Palautteen antaminen	
Mertja Jalava	17.-18.1	• Tapahtumien ja tiedotteiden haku • Henkilötietohaku • Käyttäjäkysely	
Otti Ahonen	17.-18.1	• Sisältösiivujen selaaminen ja lukeminen • Yhteiskäyttö ulkoisen järjestelmän kanssa • Sisältöihaku	
Viestintä/Sisällönsyöttäjät			
Heidi Aho	2.1 jalkkeen	• Tapahtumien ja tiedotteiden ylläpito	
Sanna Vanhatalo	2.1 jalkkeen	• Materiaalipankin käyttö • Tapahtumien ja tiedotteiden ylläpito	
Toni Enström	2.1 jalkkeen	• Materiaalipankin käyttö • I-asemalta haku • Tapahtumien ja tiedotteiden ylläpito	
Opiskelijat/Loppukäyttäjät			
Anna Raittkönen	PE 18.1 klo 8.30-16.00 Tila: Diti	• Palautteen antaminen • Oman profiilin päivittäminen • Tarkennettu haku/Sisältö • Tarkennettu haku/Henkilötieto • Sisältösiivujen selaaminen • Yhteiskäyttö ulkoisen järjestelmän kanssa	
Petri Tikka	PE 18.1 klo 8.30-16.00 Tila: Diti	• Etusiun lukeminen • Tapahtumaan ilmoittautuminen • Tapahtumien ja tiedotteiden haku • Käyttäjäkysely • I-asemalta haku • Sisäkirjautuminen	
Englanninkielisen version testaus		Tila 1347 klo 9.00-10.40	

[illegible]

Adage Usability

[illegible]



Oma profilli

[illegible]

